

AD-A039 023

SYRACUSE UNIV N Y SCHOOL OF COMPUTER AND INFORMATIO--ETC F/6 9/4
DECODING COMPLEXITY STUDY II.(U)
MAR 77 L D RUDOLPH

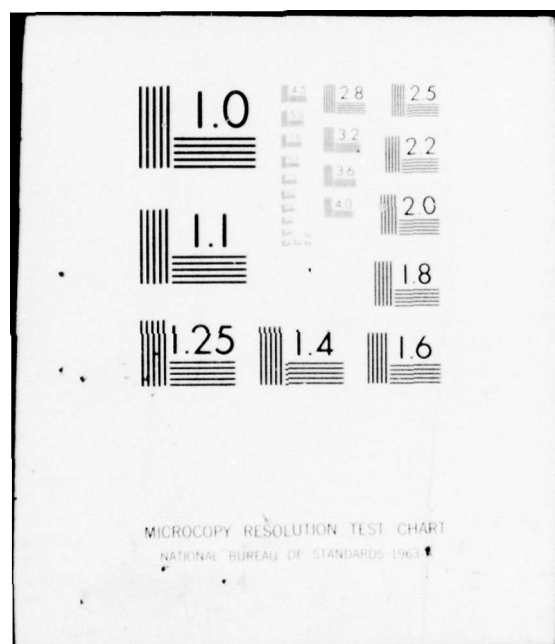
UNCLASSIFIED

RADC-TR-77-103

F30602-75-C-0121
NL

1 OF 1
AD
A039023





ADA 039023

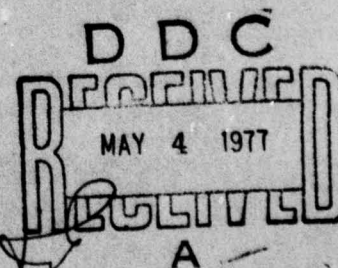
RADC-TR-77-103
Technical Report
March 1977

DECODING COMPLEXITY STUDY II
Syracuse University

12
NW



Approved for public release; distribution unlimited.



ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

ADJ NO.
DDC FILE COPY

for

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

This report has been reviewed and approved for publication.

APPROVED:

Frederick D. Schmandt

FREDERICK D. SCHMANDT
Project Engineer

APPROVED:

Fred I. Diamond

FRED I. DIAMOND
Technical Director
Communications & Control Division

DISTRIBUTION BY	
RTM	Write Section <input checked="" type="checkbox"/>
DOC	Dist Section <input type="checkbox"/>
UNARMED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION AVAILABILITY CODES	
Dist.	AVAIL. CODE SPECIAL
<i>A</i>	

FOR THE COMMANDER:

John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If this copy is not needed, return to RADC (DCLD).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-77-103	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DECODING COMPLEXITY STUDY II	5. TYPE OF REPORT & PERIOD COVERED Phase Report Jun 74 - Nov 76	
6. AUTHOR(s) Luther D. Rudolph	7. PERFORMING ORG. REPORT NUMBER N/A	
8. PERFORMING ORGANIZATION NAME AND ADDRESS Syracuse University/School of Computer & Information Science Link Hall, Syracuse NY 13210	9. CONTRACT OR GRANT NUMBER(s) F30602-75-C-0121	
10. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (DCLD) Griffiss AFB NY 13441	11. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 45192006	
12. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same	13. REPORT DATE March 1977	
14. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.	15. NUMBER OF PAGES 53	
16. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Same	17. SECURITY CLASS. (of this report) UNCLASSIFIED	
18. SUPPLEMENTARY NOTES RADC Project Engineer: Frederick D. Schmandt (DCLD)	19. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Coding Error Correction Decoding Data Transmission		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report presents the most recent results of an investigation into the complexity of decoding error-correcting codes and the development of efficient and practical decoding techniques. The most important result has been the discovery of a new general mathematical framework in which digital decoding and analog demodulation become special cases of a more general class of decoding-demodulation functions. Although we have only just begun to explore the many possibilities opened up by this discovery, results of practical importance have already been produced, including an optimum soft-decision		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

over
bpg

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Symbol-by-symbol decoding algorithm for linear codes whose complexity varies inversely with code rate.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ACKNOWLEDGEMENT

The author wishes to acknowledge the contributions to this study by his collaborator, Carlos R. P. Hartmann of Syracuse University, by Mr. Fred Schmandt of RADC (DCLD) and by Tai-Yang Hwang, Ralph Longobardi, Aileen McLoughlin, Kishan Mehrotra and Harry Schwarzlander, all of Syracuse University.

Support in this area of research is also provided by the National Science Foundation under Grant ENG75-07709.

TABLE OF CONTENTS

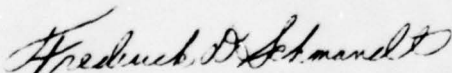
	<u>Page</u>
SECTION 1 INTRODUCTION	1
SECTION 2 ANALOG ALGEBRAIC CODING	6
SECTION 3 ANALOG THRESHOLD DECODING	13
3.1 Summary	13
3.2 The Optimum Decoding Rule	19
3.3 Asymptotic Results for the WGNC	29
SECTION 4 PARITY CHECK SET CONSTRUCTIONS	35
SECTION 5 CONCLUSIONS	43
REFERENCES	50

EVALUATION

The new general mathematical framework described in the report provides a basis for extending many existing "digital" decoding techniques into "multiplicative" decoding techniques which can be applied directly to the unquantized received word. Within the framework an optimal decoder was formulated--optimal in the sense that it provides the minimum symbol error rate possible from the received word. In practice, except for short codes, one almost always needs to back off from such a formulation to reduce complexity. Fortunately, at least in the "additive" domain, extreme reductions in complexity are often possible which do not significantly impact performance. The preliminary results obtained indicate a similar trend in the "multiplicative" domain.

The significance of the results and some possibilities for future application of the results are indicated in Section 5. Work during the remainder of the effort will focus upon obtaining the tradeoffs among performance, decoding time and hardware complexity as indicated in Section 5.

The utilization of coding in various communication applications is increasing as decoding complexity decreases. The recent Troposcatter Interleaver Contract F30602-74-C-0133 demonstrated the usefulness of coding for high speed tropo applications. An on-going contractual effort F30602-76-C-0361 titled, Demod/Decoder Integration indicates very significant performance gains are also attainable on high speed microwave line-of-sight channels. The results to date under this effort provide a basis for extending from hard decision decoders to decoders which utilize soft decisions. While additional developmental work is required, the results should be useful in the eventual development of powerful practical decoders.


FREDERICK D. SCHMANDT
Project Engineer

Section 1

INTRODUCTION

This report presents the most recent results of an investigation into the complexity of decoding error-correcting codes and the development of efficient and practical decoding techniques. For earlier results, the reader is referred to Technical Report RADC-TR-74-297, "Decoding Complexity Study", November 1974.

A major objective of this continuing research is to demonstrate that error-correcting codes are capable of providing reliable data transmission in a wide range of applications at a reasonable cost. We are convinced that the key to widespread application of coding lies in understanding and exploiting the laws that govern the trade-off between code performance and decoder complexity.

It is intuitively clear that the complexity of decoding increases ever more rapidly as the upper limit in performance is approached. Because of the steep slope of the performance-complexity curve as it approaches the performance limit, we are quite willing to suffer a small reduction in performance for the large reduction in complexity that should result. The problem is to make sure that the full reduction in complexity paid for by the loss in performance is actually obtained.

A logical approach to this problem would be to determine the optimum performance-complexity trade-off and then devise

techniques which approach or achieve this ideal. Unfortunately, we do not yet have a Shannon-type result which tells us, quantitatively, just how well we can expect to do. However, this does not prevent us from invoking general principles to deduce properties that a coding system would have to have in order to occupy a position near the theoretical performance-complexity curve. Consideration of such properties led to the formulation of the following three heuristics:

Rule 1: Do not impose any restrictions on a code beyond those necessary to obtain a given decoding advantage.

Rule 2: Make sure that the decoder fully utilizes any restrictions that are placed on the code.

Rule 3: Do not impose any restrictions on the decoder beyond those necessary to obtain a given decoding advantage.

Application of Rule 1 to the class of finite-geometry codes resulted in the development of several classes of generalized finite-geometry codes which achieve significantly improved code performance for the same decoding complexity⁽¹⁻³⁾. Application of Rule 2 to traditional majority decoding methods for cyclic codes resulted in the discovery of a new decoding algorithm which achieves the same code performance, but with a drastic reduction in decoder complexity⁽⁴⁻⁷⁾. The research carried out most recently was suggested by the application of Rule 3 to the question of whether the performance lost by hard-decision

demodulation is justified by the reduction in decoder complexity⁽⁸⁻¹⁴⁾.

In a digital communication system with one level of coding (modulation-demodulation), it is natural to design the demodulator to make hard 0-1 decisions in such a way that the probability of bit error is minimized. However, when a second level of coding (error-control encoding-decoding) is added, this demodulation strategy is no longer appropriate. In a communication system using two-level coding, the transmitted bit stream must satisfy known algebraic constraints. To make hard 0-1 decisions without regard to these constraints is to throw away information and degrade the performance of the system. This situation was tolerated for a time because it was thought that the loss in performance at the output of the demodulator was justified by the simplicity of the digital decoder that followed. This has come into question, however, and there have been many proposals for reducing this performance loss through "soft" demodulation followed by an "extended" decoder which has been modified to take advantage of the additional information provided by the demodulator.

It is natural to assume that when soft demodulation followed by an extended decoder is employed, the complexity of decoding will increase significantly. This would mean that the communication system designer would have to choose between two alternatives: (1) accept the information loss inherent in hard demodulation but use a powerful code and an efficient digital

algorithm to achieve a net performance gain, or (2) use a weak code and/or an incomplete decoding algorithm, but achieve the same net gain by eliminating the information loss at the output of the demodulator. In the spirit of questioning all assumptions about the relationship between code performance and decoder complexity, we applied heuristic Rule 3 and asked ourselves the following question: Does decoder complexity really increase drastically if we remove the restriction that the decoder be digital?

In the course of studying the previous approaches to soft-decision decoding, we became aware of a curious fact. The two best known techniques, correlation decoding of block codes and Viterbi decoding of convolutional codes, although almost always used to decode linear codes, make no essential use of the linear property. This seemed to us to be a violation of our heuristic Rule 2, so we concentrated on the question of how the algebraic structure of a linear code might be exploited in soft-decision decoding. Posing the question in this way resulted in a breakthrough to a new area of coding theory which we are now exploring. We have found that by using a new representation of finite fields, classical digital decoding techniques can be translated directly into soft-decision decoding algorithms. This strongly supports the thesis that decoder complexity does not increase drastically if we remove the restriction that the decoder be digital. Furthermore, the properties of the new algebraic framework allow consideration of new approaches to

decoding which are inapplicable in the classical digital domain.

Within this new mathematical framework, in which finite-field algebra, combinatorics and the theory of continuous functions interact in a natural way, digital decoding and analog demodulation become special cases of a more general class of decoding-demodulation functions. This means that the traditional approach of treating error-control coding as a digital add-on to the inherently analog modulation-demodulation channel may now be superseded by an integrated approach in which demodulation-decoding is viewed as a single unified signal processing function. We are sure that the ability to integrate the decoding and demodulation functions will have great impact on the design of future high-performance communication systems.

In Section 2, we discuss the new general algebraic framework which provided the context for most of the work reported herein. The major effort within this context has been the development of analog threshold decoding algorithms and the results of that effort are reported in Section 3. Section 4 presents some preliminary results of a study of parity check set construction methods for weighted majority decoding of linear block codes. Conclusions and suggestions for further research are contained in Section 5.

Section 2

ALGEBRAIC ANALOG CODING

The past twenty-five years has seen the growth of one of the most elegant and esoteric branches of applied mathematics: algebraic coding theory. Areas of mathematics previously considered to be of the utmost purity have been applied to the problem of constructing error-correcting codes and their decoding algorithms (to the point where the very concept of "pure mathematics" has become blurred⁽¹⁵⁾). Yet in spite of the impressive theoretical accomplishments, very little algebraic coding has been put into practice.

We believe that a major reason for this is that communication system designers tend to view algebraic coding as an overly fancy digital add-on to an inherently analog modulation-demodulation system, and that coding is more trouble than it is worth. Anyone who has attempted to improve the performance of an existing communication system by adding a level of error-control coding can certainly sympathize with this feeling. It is becoming increasingly clear that the best way to achieve widespread acceptance of algebraic coding is to integrate it with the modulation-demodulation system from the start.

That modulation-demodulation and encoding-decoding are simply two aspects of the overall signal design - signal processing problem is widely recognized now, and the desirability of a unified approach is apparent⁽¹⁶⁾. The modulation-

demodulation and encoding-decoding systems cannot be designed independently of one another without incurring a performance loss. The major problem occurs at the receiving end of the system when there is a mismatch between the demodulator and the decoder. The solution, clearly, is to merge the demodulation and decoding functions and design an optimum integrated decoder-demodulator. But here we run into an apples-and-oranges mathematical modelling problem.

Consider the familiar situation in which a code word of an (n,k) linear binary error-correcting code is transmitted over a time-discrete memoryless channel. We may consider the channel to be a device which adds, as vectors of real numbers, an error vector to the modulator's representation of the code word. The code word was selected from one algebraic domain, the n -dimensional vector space over the finite field $GF(2)$, and the error vector from another algebraic domain, the n -dimensional vector space over the real numbers R . An apple has been added to an orange. This poses a difficult problem at the receiver: In what domain do we process the word received at the output of the channel?

One approach is to force the error vector into the algebraic domain of the code by hard-decision demodulation. The quantized error-vector may then be viewed as a 0-1 vector which has been added, modulo 2, to the transmitted 0-1 code word, and all of the techniques of finite-field algebra, number theory and combinatorics may be employed in the design of the decoder.

We might call this the digital decoding approach. Virtually all of classical algebraic coding theory is predicated on this model. But as pointed out above, this approach is unsatisfactory from a practical point of view because of the information loss at the interface between the hard-decision demodulator and the digital decoder.

An alternative approach, which we might call probabilistic decoding, is to treat the code word as if it came from the algebraic domain of the error vector. In this case, the algebraic properties of the code (linearity, number-theoretic properties, etc.) are simply ignored. The signal processing is done entirely in the error vector domain. Two well-known examples of this approach are correlation decoding of block codes and Viterbi decoding⁽¹⁷⁾ of convolutional codes. Both methods are normally used to decode linear codes, but neither method makes any essential use of the linear property. This approach is satisfactory only for low rate or short codes.

A third approach is to attempt to exploit both algebraic domains by combining digital and probabilistic decoding. Examples of such hybrid decoding schemes are: Wagner decoding⁽¹⁸⁾, generalized-minimum-distance decoding⁽¹⁹⁾, weighted-erasure decoding⁽²⁰⁾ and decoding with channel-measurement information⁽²¹⁾. Although these schemes show improvement over strictly digital or strictly probabilistic decoding in many instances, one gets the impression that the apples-and-oranges problem remains unresolved.

As a result of surveying the existing decoding techniques,

it occurred to us to ask whether anything could be done about the apparent incompatibility between these two algebraic domains. Clearly, nothing can be done about Mother Nature's error vector domain, but what about the man-made algebraic domain of the code? This line of inquiry led to the discovery of a more general algebraic domain in which the analog error vector and the digital code word co-exist in a natural way. This is achieved through the use of a new representation of finite fields which we will now describe. For simplicity, we restrict our discussion to fields of prime order. The extension to fields of prime-power order is straightforward.

The finite field of p elements, $GF(p)$, is usually represented by the ring of integers mod p . We will call this the "additive representation" of $GF(p)$ and denote it by $S = \langle S, \oplus, \odot \rangle$ where $S = \{0, 1, \dots, p-1\}$, and " \oplus " and " \odot " are modulo p addition and multiplication. The new representation of $GF(p)$, which we call the "multiplicative representation", will be denoted by $S' = \langle S', \cdot, * \rangle$ where $S' = \{1, \alpha, \alpha^2, \dots, \alpha^{p-1}\}$ is the set of complex p^{th} roots of unity, " \cdot " is ordinary multiplication of complex numbers, and " $*$ " is a new operation defined by

$$u * v = v^{\log_{\alpha} u},$$

where the principal value of the logarithm is taken. To show that S' is indeed a representation of $GF(p)$, it is necessary only to establish the existence of an isomorphism from S to S' . Thus let f be any function from S to the complex numbers C such that for all $i \in S$, $f(i) = \alpha^i$. It is easy to verify that f is

such an isomorphism.

The important point is that the operations "." and "*" of the multiplicative representation of $GF(p)$ are defined for all nonzero elements of C , not just the p^{th} roots of unity.¹ We have thus constructed a general algebraic system $\langle C, +, \cdot, * \rangle$ which contains the multiplicative representation of $GF(p)$. Every algebraic equation that can be written in the classical S -domain can be translated directly into an equivalent equation in the S' -domain. But once in the S' -domain, the algebraic equation extends immediately to non-digital arguments (i.e. arguments which are not restricted to be p^{th} roots of unity). This will be discussed in the next section, but we can give a simple explanation here. In conventional digital decoding of a linear (n,k) code, a parity check is defined by

$$s_i \equiv \sum_{j=1}^n h_{ij} r_j \pmod{p}$$

where (r_1, \dots, r_n) is the received word and (h_{i1}, \dots, h_{in}) is a word in the dual code. It is assumed here that $r_i \in S$, for, if not, the algebraic operations are not defined. The corresponding equation in the S' domain is

$$s'_i = \sum_{j=1}^n h'_{ij} * r'_j = \sum_{j=1}^n r'_j \cdot h_{ij}$$

¹There is a technical difficulty with the definition of z^x when we allow z to be complex if we require that all of the usual laws of exponents hold (which in our development we do not). For a discussion of this point, see A. M. Gleason, "Fundamentals of Abstract Analysis," pp. 324-326, Addison-Wesley 1966.

(where $h'_{ij} = \alpha^{h_{ij}}$, etc.). If the r'_j are p^{th} roots of unity, then there is nothing new. But the r'_j need not be so restricted. The algebraic operations are still defined when the r'_j are any nonzero complex numbers. This means that any conventional "additive" digital decoder can be translated into a corresponding "multiplicative" decoder and then applied directly to the raw, unquantized received word. The question, of course, is: how well does this work? As will be seen in the next section, it can work surprisingly well, but we know far too little at this time to make any more specific statements. We are in the somewhat embarrassing position of having a variety of new demodulation-decoding techniques which work remarkably well, but which we understand only marginally. We are currently investigating such problems as interpreting the meaning of the syndrome when the digital restriction is removed, and determining how many "multiplicative" parity checks are required to specify a code in this more general domain. We are studying the interaction of probabilistic, algebraic and combinatorial mechanisms in an effort to find the proper viewpoint from which to make sense of it all. To date, the best insights have come through the use of abstract harmonic analysis (group characters, finite Fourier transforms, etc.). At this point we know that all of classical algebraic coding theory can be translated from the additive domain into the multiplicative domain, and that once in this new domain a bewildering number of analog

processing extensions become possible. The extension which we are exploring currently is discussed in Section 3.

Section 3

ANALOG THRESHOLD DECODING

3.1 Introduction

Majority logic decoding and the more general threshold decoding constitute widely studied areas of algebraic coding theory^(4-7,22-40). Majority decoding usually takes the form of a symbol-by-symbol decoding algorithm for linear block or convolutional codes. Most majority algorithms make strong use of both the linearity of the code and any special combinatorial structure the code may have. Because of the principal investigators' familiarity with the area, majority decoding was the first technique to be translated into the multiplicative domain and extended to analog processing of the unquantized received word. (Actually, the discovery of the multiplicative extension of majority decoding^(6,8) predated the discovery of the general multiplicative algebraic domain.) Following Massey⁽²⁹⁾, we call this extended decoding method "analog threshold decoding".

In the previous section, we pointed out that any function f which, for all $i \in S$, maps $i \rightarrow \alpha^i$ is an isomorphism from S , the additive representation of $GF(p)$, to S' , the multiplicative representation of $GF(p)$. One way to convert an additive digital decoder to a multiplicative analog decoder is to physically implement the function f and apply it to the output of the channel. Our first experiment along these lines involved a majority decoder for a linear binary block code transmitted

over a time-discrete memoryless channel. The initial choice for the isomorphism f was $f(x) = \cos \pi x$. (In the binary case, $\alpha = -1$, $S = \{0, 1\}$ and $S' = \{1, -1\}$).

In conventional one-step majority decoding, the unquantized received word (r_1, \dots, r_n) is converted to a 0-1 vector (u_1, \dots, u_n) by hard-decision demodulation, various modulo 2 parity check sums involving the $\{u_i\}$ are computed and an estimate of c_i , the i^{th} transmitted code digit, is obtained by majority decision on these sums. In analog threshold decoding, the received word (r_1, \dots, r_n) is converted to a real-valued vector $(\cos \pi r_1, \dots, \cos \pi r_n)$, the corresponding parity check products involving the $\{\cos \pi r_i\}$ are computed, and the estimate of c_i is obtained by thresholding on the sum of these products.

The reader might well ask at this point why anyone would choose the function $\cos \pi$ to be the isomorphism from S to S' . A periodic "soft-decision" demodulation function hardly makes sense from a communication system designer's point of view. The reasons for this choice are of historical interest only and it is certainly true that $\cos \pi$ would never be used in practice. However, it is also true that $\cos \pi$ works surprisingly well, and that it is a convenient function to work with from a theoretician's point of view (i.e., theorems can be proved).

In order to talk about the performance of an analog threshold decoder, we have to define error-correcting capability over the real numbers \mathbb{R} . The natural distance measure over \mathbb{R}^n is the Euclidean metric, and it is easily

verified that a binary (n,k) code with minimum Hamming distance d_H has minimum Euclidean distance $d_E = \sqrt{d_H}$. We say that a decoding function is a nearest-neighbor decoding rule if it maps a received vector onto a nearest word in the code, and a radius- r decoding rule if it maps a received vector onto a nearest word in the code whenever the vector is within Euclidean distance r of a code word. The maximum radius possible without having overlapping spheres is $r = \frac{d_E}{2}$, and a decoding function which achieves this radius is called a maximum-radius decoding rule. (Note the obvious analogy with t -error-correction in digital decoding.)

Using the demodulation function $\cos\pi$, we were able to prove the following results for linear binary block codes⁽⁸⁾. First, a one-step orthogonalizable code of minimum Hamming distance d_H can be maximum-likelihood decoded by a one-step analog threshold decoder using at most d_H parity check products. Second, a Hamming code of length $n = 2^m - 1$ can be maximum-radius decoded using 2^{m-1} products. Finally, any L -step orthogonalizable code with minimum Hamming distance d_H can be maximum-radius decoded by a sequential code reduction decoder⁽⁴⁾ whose first stage is an analog threshold decoder using d_H products and whose remaining stages are digital, provided that the subcodes used in decoding are all capable of correcting $d_H - 1$ or fewer digital errors (which is almost always the case). We also showed that maximum-radius decoding - however achieved - is asymptotically optimum for the white Gaussian channel. More

recently, we have been able to extend our earlier results and show that whenever it is possible to find a set of R parity checks which all check the i^{th} position, but no more than λ of which check any other position, then a one-step analog threshold decoder will do radius- r decoding with $r = \frac{1}{2} \sqrt{\frac{R+\lambda}{\lambda}}$.

(For orthogonalizable codes, $R = d_H - 1$ and $r = \frac{1}{2} \sqrt{d_H} = \frac{d_H}{2}$.)

Since $f = \cos \pi$ was clearly not an optimum choice, we experimented with other functions in an effort to understand what makes a good demodulation function. We ran computer simulations for the white Gaussian channel using a variety of functions - including $\cos \pi$ - with inconclusive results. The analog threshold decoders consistently outperformed the corresponding digital majority decoders, but a particular function would be better for one code than for another, or would perform better at one signal-to-noise ratio (SNR) than at another. It was not until we began to consider the possibility of adaptive analog threshold decoding that the optimum function was found.

In the case of a linear binary (n, k) code, the optimum soft-decision function (optimum in the sense that the probability of bit error is minimized over any time-discrete memoryless channel when the code words are equiprobable) is

$$f(x) = \frac{1 - \phi(x)}{1 + \phi(x)}$$

where $\phi(x) = \text{Pr}(x|1)/\text{Pr}(x|0)$ is the likelihood ratio. This function is optimum when (1) all 2^{n-k} parity check products are used and (2) the products are weighted equally. We have been

able to generalize this result to any linear block or convolutional code over $GF(q)$ ⁽¹³⁾.

The reader may have noticed that the function $f(x)$ as given is not an isomorphism from S to S' . We could easily make it so by normalizing, but then the weights assigned to the parity products would become functions of the SNR. When the function as given is used, the contributions of the various parity products are automatically scaled according to their reliability at the SNR on the channel. At high SNR, the 2^{n-k} parity products contribute more-or-less equally, while at low SNR the only significant contributions are from parity products which correspond to minimum weight words in the dual code. It is interesting to note that for the white Gaussian channel, the optimum function approaches a $[-1,+1]$ step function as $SNR \rightarrow \infty$. Analog threshold decoding would be mathematically equivalent to digital majority decoding if the step function were actually used. This illustrates very nicely the fact that digital decoding is a special limiting case of this more general class of decoding - demodulation functions.

The discovery of the optimum soft-decision symbol-by-symbol analog threshold decoding algorithm is significant because the complexity of the algorithm varies with the size of the dual code and is thus inversely related to code rate. This decoding method therefore is to high-rate codes what correlation and Viterbi decoding are to low rate codes, which fills an important gap in the arsenal of decoding techniques. But even more

significant, perhaps, is the concept of soft-decision decoding in the dual code domain itself. In classical code domain decoding, there is no graceful way to give up a small amount of performance in order to reduce the complexity of the decoder. For example, one cannot discard half of the matched filters in a correlation receiver, or half of the microcomputers in a Viterbi decoder. The effect on performance would be disastrous. This is not so in the case of dual code domain decoding. If we were to throw away, at random, half of the parity check products in an optimum analog threshold decoder, we would not expect a significant loss in performance. The reason for this is that the dual code domain expansion of the decoding function is essentially a Fourier series, and even a fairly severe truncation of the series should result in no more than a small overall degradation of performance, the loss being independent of the code word transmitted. To support this view, we cite the results of some very recent simulations carried out by CNR, Inc.⁽⁴¹⁾ for the (21,11) code on the white Gaussian channel. Reducing the number of parity products from 1024 to 6 resulted in a loss of less than 1 db at the bit error rate of 2×10^{-3} .

Much remains to be done in this area, particularly on the problem of suboptimum analog threshold decoding. We still do not know what the optimum demodulation function is when a proper subset of the available parity checks is used, or even if the optimum function can be factored into an adaptive part, which is a function of the SNR, and a fixed part which is a

function of the set of parity check products to be used. However, the preliminary findings are certainly encouraging and we expect that this line of investigation will continue to produce results of theoretical and practical importance.

We now present, in detail, the new optimum symbol-by-symbol decoding rule for linear codes.

3.2 The Optimum Decoding Rule

For convenience, we present the decoding rule for linear block codes. The extension to convolutional codes is immediate and will be obvious from the examples.

Let $\underline{c} = (c_0, c_1, \dots, c_{n-1})$ denote any code word of an (n, k) linear block code C over $GF(p)$ and $\underline{c}'_j = (c'_{j0}, c'_{j1}, \dots, c'_{j, n-1})$ the j th code word of the $(n, n-k)$ dual code C' . A code word \underline{c} is transmitted over a time-discrete memoryless channel with output alphabet B . The received word is denoted by $\underline{r} = (r_0, r_1, \dots, r_{n-1})$, $r_j \in B$. The decoding problem is: given \underline{r} , compute an estimate \hat{c}_m of the transmitted code symbol c_m in such a way that the probability that \hat{c}_m equals c_m is maximized. Other notation: $\omega \equiv \exp[2\pi\sqrt{-1}/p]$ (primitive complex p th root of unity); $\delta_{ij} = 1$ if $i = j$ and 0 otherwise; $\Pr(x)$ is the probability of x and $\Pr(x|y)$ is the probability of x given y . Unless otherwise stated, the elements of $GF(p)$ are taken to be the integers $0, 1, \dots, p-1$ and all arithmetic operations are performed in the field of complex numbers.

DECODING RULE:

Set $\hat{c}_m = s$, where $s \in GF(p)$ maximizes

the expression

$$A_m(s) = \sum_{t=0}^{p-1} \omega^{-st} p^{n-k} \sum_{j=1} \left[\prod_{\ell=0}^{n-1} \sum_{i=0}^{p-1} \omega^{-i(c'_{j\ell} - t\delta_{m\ell})} \Pr(r_\ell | i) \right]. \quad (1)$$

Theorem: Decoding Rule (1) maximizes the probability that

$$\hat{c}_m = c_m.$$

(Proof) We must show that choosing s to maximize $A_m(s)$ is equivalent to maximizing the probability that c_m equals s given the received word \underline{r} . We do this directly by showing that $\Pr(c_m = s | \underline{r}) = \lambda A_m(s)$, where λ is a positive constant which is independent of s . We first note that

$$\begin{aligned} \Pr(c_m = s | \underline{r}) &= \sum_{\underline{c} \in C, c_m = s} \Pr(\underline{c} | \underline{r}) \\ &= \sum_{\underline{c} \in C, c_m = s} \Pr(\underline{r} | \underline{c}) [\Pr(\underline{c}) / \Pr(\underline{r})]. \end{aligned} \quad (2)$$

Since the code words of C are equiprobable, $\Pr(\underline{c}) = p^{-k}$ and

(2) becomes

$$\Pr(c_m = s | \underline{r}) = [p^{-k} / \Pr(\underline{r})] \sum_{\underline{c} \in C} \Pr(\underline{r} | \underline{c}) \delta_{0, (\underline{c} \cdot \underline{e}_m - s)}, \quad (3)$$

where $\underline{e}_m = (\delta_{m0}, \delta_{m1}, \dots, \delta_{m, (n-1)})$ is the vector with 1 in the m^{th} position and 0 elsewhere. In terms of their finite Fourier transforms,

$$\delta_{0, (\underline{c} \cdot \underline{e}_m - s)} = p^{-1} \sum_{t=0}^{p-1} \omega^{t(\underline{c} \cdot \underline{e}_m - s)} \quad (4)$$

$$P_r(\underline{r}|\underline{c}) = p^{-n} \sum_{\underline{u} \in V_n} F(\underline{r}, \underline{u}) \omega^{\underline{u} \cdot \underline{c}} \quad (5)$$

where

$$F(\underline{r}, \underline{u}) = \sum_{\underline{v} \in V_n} \Pr(\underline{r}|\underline{v}) \omega^{-\underline{u} \cdot \underline{v}}, \quad (6)$$

$\underline{u} = (u_0, u_1, \dots, u_{n-1})$ and $\underline{v} = (v_0, v_1, \dots, v_{n-1})$ any elements of V_n , the vector space of all n -tuples over $GF(p)$. Substituting (4) and (5) in (3) yields

$$\begin{aligned} \Pr(c_m = s | \underline{r}) &= [p^{-n-k-1} / \Pr(\underline{r})] \sum_{\underline{c} \in C} \left[\sum_{\underline{u} \in V_n} F(\underline{r}, \underline{u}) \omega^{\underline{u} \cdot \underline{c}} \right] \left[\sum_{t=0}^{p-1} \omega^{t(\underline{c} \cdot \underline{e}_m - s)} \right] \\ &= [p^{-n-k-1} / \Pr(\underline{r})] \sum_{t=0}^{p-1} \omega^{-st} \sum_{\underline{u} \in V_n} F(\underline{r}, \underline{u}) \left[\sum_{\underline{c} \in C} \omega^{(\underline{u} + t\underline{e}_m) \cdot \underline{c}} \right]. \quad (7) \end{aligned}$$

By the orthogonality properties of group characters, we know that

$$\sum_{\underline{c} \in C} \omega^{\underline{v} \cdot \underline{c}} = \begin{cases} p^k & \text{if } \underline{v} \in C' \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Applying (8) to (7) gives

$$\Pr(c_m = s | \underline{r}) = [p^{-n-1} / \Pr(\underline{r})] \sum_{t=0}^{p-1} \omega^{-st} p^{n-k} \sum_{j=1}^{n-k} F(\underline{r}, \underline{c}'_j - t\underline{e}_m). \quad (9)$$

Since the channel is memoryless, we may write (6) as

$$F(\underline{r}, \underline{u}) = \sum_{\underline{v} \in V_n} \prod_{\ell=0}^{n-1} \Pr(r_\ell | v_\ell) \omega^{-\underline{u} \cdot \underline{v}} = \prod_{\ell=0}^{n-1} \sum_{i=0}^{p-1} \Pr(r_\ell | i) \omega^{-iu_\ell}. \quad (10)$$

Substituting (10) in (9) yields

$$\Pr(c_m = s | \underline{r}) = [p^{-n-1} / \Pr(\underline{r})] \sum_{t=0}^{p-1} \omega^{-st} p^{n-k} \sum_{j=1}^{p-1} \left[\prod_{\ell=0}^{n-1} \sum_{i=0}^{p-1} \omega^{-i(c'_{j\ell} - t\delta_{m\ell})} \right] \Pr(r_\ell | i) = [p^{-n-1} / \Pr(\underline{r})] A_m(s) . \quad \text{Q.E.D.}$$

As one might expect, the decoding rule takes a comparatively simple form in the binary case: set $\hat{c}_m = 0$ if $A_m(0) > A_m(1)$ and $\hat{c}_m = 1$ otherwise. It is more convenient however to state the rule in terms of the likelihood ratio

$$\phi_m = \Pr(r_m | 1) / \Pr(r_m | 0) .$$

Substituting (1) into the inequality $A_m(0) > A_m(1)$ yields

$$\sum_{t=0}^1 2^{n-k} \sum_{j=1}^{n-1} \prod_{\ell=0}^1 \sum_{i=0}^1 (-1)^{-i(c'_{j\ell} - t\delta_{m\ell})} \Pr(r_\ell | i) >$$

$$\sum_{t=0}^1 (-1)^t 2^{n-k} \sum_{j=1}^{n-1} \prod_{\ell=0}^1 \sum_{i=0}^1 (-1)^{-i(c'_{j\ell} - t\delta_{m\ell})} \Pr(r_\ell | i) ,$$

or

$$2^{n-k} \sum_{j=1}^{n-1} \prod_{\ell=0}^1 \left[\Pr(r_\ell | 0) + (-1)^{-c'_{j\ell} - \delta_{m\ell}} \Pr(r_\ell | 1) \right] > 0 . \quad (11)$$

Dividing both sides of (11) by $\prod_{\ell=0}^{n-1} \Pr(r_\ell | 0)$ and using the definition of the likelihood ratio, we have

$$2^{n-k} \sum_{j=1}^{n-1} \prod_{\ell=0}^1 \left[1 + \phi_\ell (-1)^{-c'_{j\ell} - \delta_{m\ell}} \right] > 0 . \quad (12)$$

Then dividing both sides of (12) by the positive quantity

$$\prod_{\ell=0}^{n-1} [1 + \phi_\ell] ,$$

$$2^{n-k} \sum_{j=1}^{n-1} \prod_{\ell=0}^{n-1} \frac{1+\phi_{\ell}(-1)^{-c'_{j\ell}-\delta_{m\ell}}}{1+\phi_{\ell}} > 0 .$$

Finally, using the identity

$$\frac{1+\phi_{\ell}(-1)^{-c'_{j\ell}-\delta_{m\ell}}}{1+\phi_{\ell}} = \left(\frac{1-\phi_{\ell}}{1+\phi_{\ell}} \right)^{c'_{j\ell} \oplus \delta_{m\ell}}$$

where ' \oplus ' denotes modulo 2 addition, we obtain the

BINARY DECODING RULE:

Set $\hat{c}_m = 0$ if

$$2^{n-k} \sum_{j=1}^{n-1} \prod_{\ell=0}^{n-1} \left(\frac{1-\phi_{\ell}}{1+\phi_{\ell}} \right)^{c'_{j\ell} \oplus \delta_{m\ell}} > 0 \quad (13)$$

and $\hat{c}_m = 1$ otherwise.

We remark that up to this point we have ignored the question of how one retrieves the decoded information symbols from the code word estimate $\hat{\underline{c}}$. This could be a problem because, when a symbol-by-symbol decoding rule is used, $\hat{\underline{c}}$ is not in general a code word. In the case of block codes, we could insist that the code be systematic without loss of generality, but there might be some objection to this restriction in the case of convolutional codes. As it turns out, this is not a problem since the decoding rule is easily modified to produce estimates of the information symbols directly if need be. Simply note that every information symbol a_m can be expressed as a linear combination, over $GF(p)$, of code words symbols c_m ,

i.e. $a_m = \sum_{\ell} b_{m\ell} c_{\ell}$, $b_{m\ell} \in GF(p)$, and that the proof of the theorem goes through intact if we substitute $\sum_{\ell} b_{m\ell} c_{\ell}$ for \hat{c}_m and $b_{m\ell}$ for $\delta_{m\ell}$ in (1).

EXAMPLES:

(a) (7,4) Hamming code

We will illustrate the decoding rule for the received symbol r_0 . Since the (7,4) code is cyclic, r_1, \dots, r_6 may be decoded simply by cyclically permuting the received word r in the buffer store.

The Binary Decoding Rule (13) in this case becomes

$$\hat{c}_0 = 0 \text{ iff } \sum_{j=1}^8 \prod_{\ell=0}^6 \left(\frac{1-\phi_{\ell}}{1+\phi_{\ell}} \right)^{c'_j \oplus \delta_0} > 0. \quad (14)$$

The parity check matrix H of the (7,4) code and its row space C' are shown below.

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \quad \begin{matrix} (a) \\ (b) \\ (c) \end{matrix}$$

	c_0	c_1	c_2	c_3	c_4	c_5	c_6	
C' :	0	1	1	1	0	1	0	(a)
	1	0	0	1	1	1	0	(a⊕b)
	0	0	1	1	1	0	1	(c)
	1	1	0	1	0	0	1	(a⊕c)
	0	1	0	0	1	1	1	(b⊕c)
	1	0	1	0	0	1	1	(a⊕b⊕c)

(15)

Let $\rho_{\ell} = (1-\phi_{\ell})/(1+\phi_{\ell})$. Then substituting (15) into (14) gives

$$\hat{c}_0 = 0 \text{ iff } \rho_0 + \rho_1\rho_2\rho_4 + \rho_2\rho_5\rho_6 + \rho_1\rho_3\rho_6 + \rho_3\rho_4\rho_5 + \rho_0\rho_1\rho_2\rho_3\rho_5 + \rho_0\rho_2\rho_3\rho_4\rho_6 + \rho_0\rho_1\rho_4\rho_5\rho_6 > 0. \quad (16)$$

The decoder configuration corresponding to (16) is shown in Figure 1.

The reader will probably recognize the similarity between the decoder of Figure 1 and a one-step majority decoder using non-orthogonal parity checks⁽³¹⁾. And in fact if the "soft decision" function $(1-\phi(x))/(1+\phi(x))$ were replaced by the "hard decision" function $f(x) = -1$ if $x > \frac{1}{2}$ and $+1$ otherwise, and if the last three parity checks in the decoder were deleted, then the resulting circuit would be mathematically equivalent to a conventional one-step majority decoder. Parity checks in the circuit of Figure 1 would be computed by taking products of $+1$'s and -1 's, rather than by taking modulo 2 sums of 0 's and 1 's as would be the case in a conventional digital decoding circuit.

(b) (4,3,3) convolutional code

We now illustrate the decoding rule for the received symbol r_0 using an $(n_0, k_0, m) = (4, 3, 3)$ convolutional code (from Peterson and Weldon⁽⁴²⁾, page 395).

The Binary Decoding Rule (13) in this case becomes

$$\hat{c}_0 = 0 \text{ iff } \sum_{j=1}^{\infty} \prod_{\ell=0}^{\infty} \left(\frac{1-\phi_{\ell}}{1+\phi_{\ell}} \right)^{c'_{j\ell} \oplus \delta_{0\ell}} > 0. \quad (17)$$

Of course, there are only a finite number of nonzero terms in (17), the number depending upon the length of the transmitted code sequence. The initial portions of the parity check matrix H of the $(4, 3, 3)$ code and its row space C' are shown below.

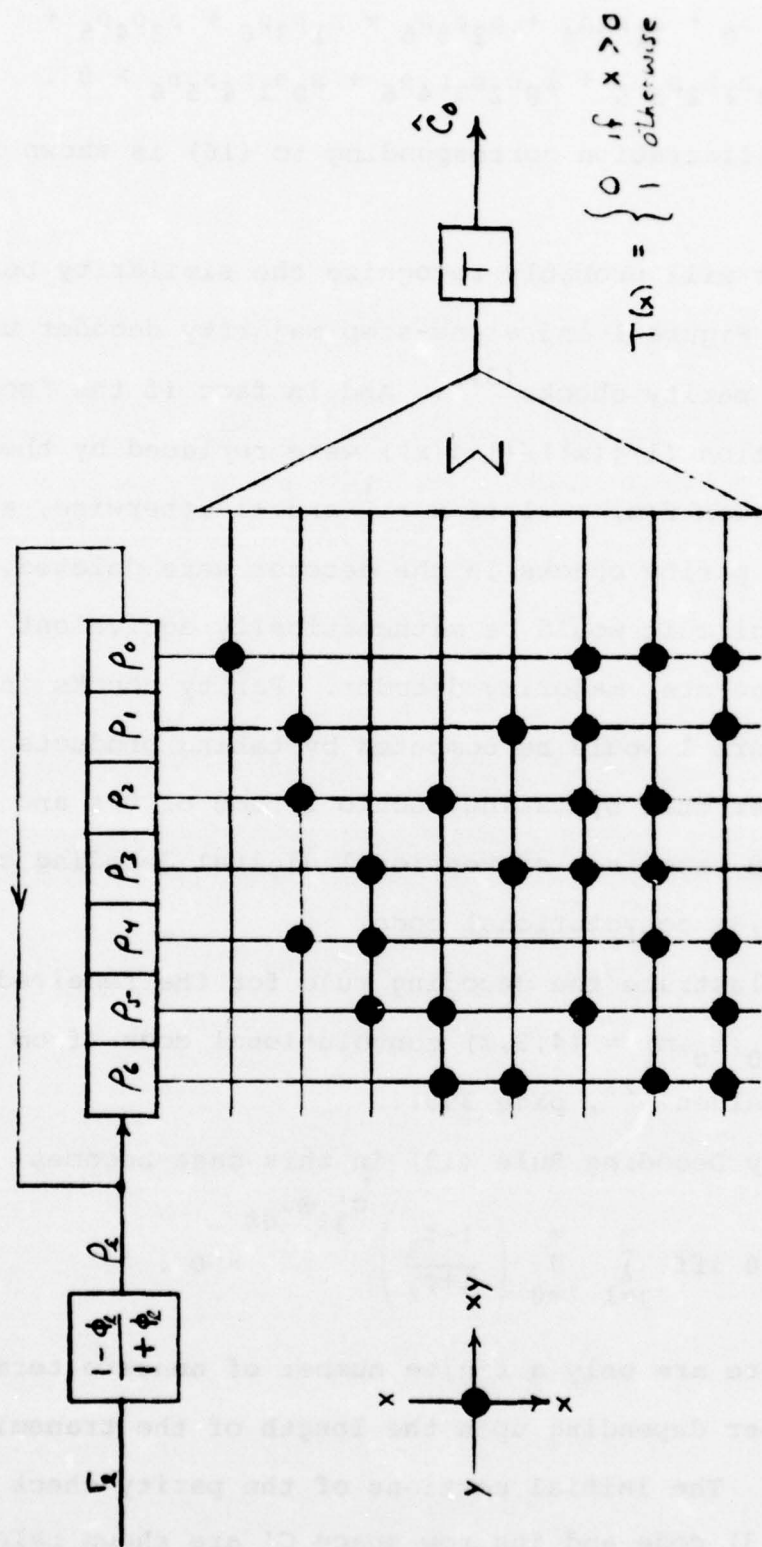


Figure 1. Decoder for the (7,4) code.

$$\begin{array}{rcl}
H = & \begin{array}{l} \boxed{\begin{array}{l} 1 \ 1 \ 1 \ 1 \ 0 \ \dots \\ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ \dots \\ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ \dots \\ \vdots \end{array}} & \begin{array}{l} (a) \\ (b) \\ (c) \\ \vdots \end{array} \\
& 0 \ \dots & \\
& \begin{array}{l} 1 \ 1 \ 1 \ 1 \ 0 \ \dots \\ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ \dots \\ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ \dots \\ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ \dots \\ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ \dots \\ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ \dots \\ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ \dots \\ \vdots \end{array} & \begin{array}{l} (a) \\ (b) \\ (a \oplus b) \\ (c) \\ (a \oplus c) \\ (b \oplus c) \\ (a \oplus b \oplus c) \\ \vdots \end{array} \\
C': & & (18)
\end{array}$$

As before, let $\rho_\ell = (1 - \phi_\ell) / (1 + \phi_\ell)$. Then substituting (18) into (17) gives

$$\begin{aligned}
\hat{c}_0 = 0 \text{ iff } & p_0 + \rho_1 \rho_2 \rho_3 + \rho_2 \rho_4 \rho_5 \rho_6 \rho_7 + \\
& + \rho_0 \rho_1 \rho_3 \rho_4 \rho_5 \rho_6 \rho_7 + \dots > 0 .
\end{aligned} \tag{19}$$

The decoding diagram corresponding to (19) is shown in Figure 2. This takes the form of a trellis diagram for the (4,1,3) dual code C' with the c'_{j0} positions in the branch labels complemented. (In general, to decode r_m the c'_{jm} positions would be complemented.) Note that the all-zero state acts as the accumulator for the terms of (19).

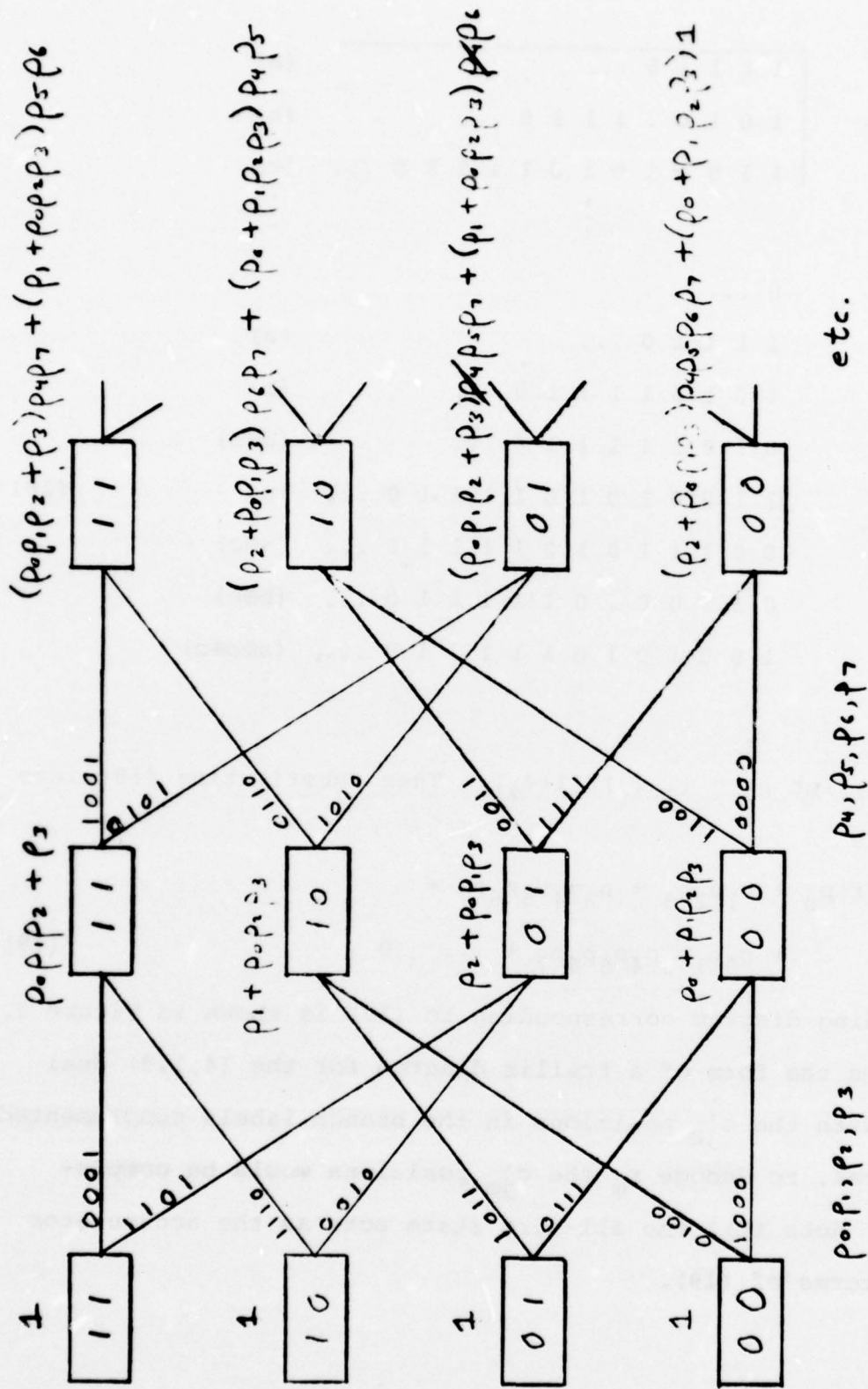


Figure 2. Decoder for the (4,3,3) code.

Since a different storage unit must be used for each symbol to be decoded, the amount of storage for this type of decoder grows linearly with the length of the transmitted code sequence. This is also true of a Viterbi decoder, which must keep track of its path-elimination decisions.

We now restrict our attention to linear binary block codes with equiprobable code words transmitted over the additive white Gaussian noise channel by antipodal signalling and present asymptotic expressions for the probability of bit error, P_{BIT} , for both high and low SNR.

3.3 Asymptotic Results for the WGNC

The optimum bit-by-bit decoding rule for the m^{th} digit of an (n,k) linear binary block code C is: set $\hat{c}_m = s$, where $s \in \text{GF}(2)$ maximizes $P(c_m = s | \underline{r})$. Here $\underline{c} = (c_0, \dots, c_{n-1})$ is the transmitted code word, $\underline{r} = (r_0, \dots, r_{n-1})$ is the received word, and \hat{c}_m is the decoder's estimate of the transmitted code digit c_m . The probability of bit error is then given by

$$P_{\text{BIT}} = P[P(\hat{c}_m = c_m | \underline{r}) \leq P(\hat{c}_m \neq c_m | \underline{r})] .$$

The derivation of our results is simplified by assuming that the all-0 code word is transmitted. It is easily seen that the assumption is valid for the case under consideration because of the group property of the code, which renders the view of n -space from one code word the same as from another, and because of the symmetry of the noise, from which it follows that noise vector $\underline{e} = (e_0, \dots, e_{n-1})$ occurs when $\underline{c} = (c_0, \dots, c_{n-1})$ is trans-

mitted with the same probability that $\underline{e}' = (-1^{c_0} e_0, \dots, -1^{c_{n-1}} e_{n-1})$ occurs when the all-0 code word is transmitted.

When the all-0 code word is transmitted, the m^{th} component of the received word is

$$r_m = \sqrt{E} + e_m ,$$

where E is the signal energy per channel bit and e_m is a noise sample of a Gaussian process with single-sided noise power per hertz N_0 . The variance of e_m is $N_0/2$ and the SNR for this channel is $\gamma = E/N_0$. In order to account for the redundancy in codes of different rates, we will use the SNR per transmitted bit of information, $\gamma_b = E_b/N_0 = \gamma n/k = \gamma/R$, in our derivations. In terms of γ_b , we can write the likelihood ratio as

$$\begin{aligned} \phi(r_m) &= \exp\left(-\frac{(r_m + \sqrt{E})^2}{N_0}\right) / \exp\left(-\frac{(r_m - \sqrt{E})^2}{N_0}\right) \\ &= \exp(-4r_m \sqrt{\gamma/N_0}) = \exp(-4r_m \sqrt{R\gamma_b/N_0}) . \end{aligned} \quad (1)$$

The m^{th} component of the receive word \underline{r} will be decoded incorrectly if and only if

$$P(c_m = 0 | \underline{r}) \leq P(c_m = 1 | \underline{r}) ,$$

where $\underline{r} = (\sqrt{E} + e_0, \dots, \sqrt{E} + e_{n-1})$. In other words,

$$P_{\text{BIT}} = P\left[\bigcup_{\underline{c} \in S_0} P(\underline{c} | \underline{r}) \leq \bigcup_{\underline{c} \in S_1} P(\underline{c} | \underline{r})\right] , \quad (2)$$

where $S_i = \{\underline{c} \in C | c_m = i\}$, $i = 0, 1$. Since the channel is memoryless, and the code words are equiprobable (so that we may invoke Bayes' formula), (2) may be written as

$$P_{\text{BIT}} = P \left[\sum_{c \in S_0} \prod_{\ell=0}^{n-1} P(r_\ell | c_\ell) \leq \sum_{c \in S_1} \prod_{\ell=0}^{n-1} P(r_\ell | c_\ell) \right]. \quad (3)$$

Since $\prod_{\ell=0}^{n-1} P(r_\ell | 0) > 0$, we can rewrite (3) in terms of the likelihood ratio as

$$P_{\text{BIT}} = P \left[\sum_{c \in S_0} \prod_{\ell=0}^{n-1} \phi(r_\ell)^{c_\ell} \leq \sum_{c \in S_1} \prod_{\ell=0}^{n-1} \phi(r_\ell)^{c_\ell} \right]. \quad (4)$$

Substituting (1) into (4) yields

$$P_{\text{BIT}} = P \left[\sum_{c \in S_0} \exp(-r \cdot c \cdot 4\sqrt{R\gamma_b/N_0}) \leq \sum_{c \in S_1} \exp(-r \cdot c \cdot 4\sqrt{R\gamma_b/N_0}) \right]. \quad (5)$$

We now fix N_0 and vary E_b to obtain the asymptotic behavior of P_{BIT} as γ_b decreases (denoted by A.B. $(P_{\text{BIT}})_{\gamma_b \rightarrow 0}$) and as γ_b increases (denoted by A.B. $(P_{\text{BIT}})_{\gamma_b \rightarrow \infty}$).

Low SNR Case

For x small, $\exp x \approx 1 + x$, so for γ_b in a small neighborhood of zero we may write (5) as

$$\text{A.B.}_{\gamma_b \rightarrow 0} (P_{\text{BIT}}) \approx P \left[\sum_{c \in S_0} (1 - r \cdot c \cdot 4\sqrt{R\gamma_b/N_0}) \leq \sum_{c \in S_1} (1 - r \cdot c \cdot 4\sqrt{R\gamma_b/N_0}) \right]. \quad (6)$$

By [42, problem 3.5], (6) can then be written as

$$\text{A.B.}_{\gamma_b \rightarrow 0} (P_{\text{BIT}}) \approx P \left[\sum_{c \in S_0} -r \cdot c \cdot 4\sqrt{R\gamma_b/N_0} \leq \sum_{c \in S_1} -r \cdot c \cdot 4\sqrt{R\gamma_b/N_0} \right],$$

which implies that

$$\text{A.B.}_{\gamma_b \rightarrow 0} (P_{\text{BIT}}) \approx P \left[\sum_{c \in S_0} r \cdot c \geq \sum_{c \in S_1} r \cdot c \right]. \quad (7)$$

S_0 is a linear binary $(n, k-1)$ code, so if the vectors of S_0 are arranged as rows of a matrix M_0 , then each column will contain all 0's or else 2^{k-2} 0's and 2^{k-2} 1's. Then if we arrange, as rows of a matrix M_1 , the vectors of set S_1 , the columns of M_1 which correspond to all-0 columns of M_0 must contain all 1's, and all other columns of M_1 will contain 2^{k-2} 0's and 2^{k-2} 1's. Using this fact, we can write (7) as

$$\text{A.B. } (P_{\text{BIT}})_{\gamma_b \rightarrow 0} \approx P \left[\sum_{\ell=j_1}^{j_\theta} r_\ell \leq 0 \right],$$

where j_1, \dots, j_θ are the columns of M_0 which contain all 0's. Since r_ℓ is a normally distributed random variable with mean \sqrt{E} and variance $N_0/2$, and since the noise is white,

$$P \left[\sum_{\ell=j_1}^{j_\theta} r_\ell \leq 0 \right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(-x^2/2) dx = Q(\sqrt{2\theta R \gamma_b}).$$

The desired asymptotic expression for low SNR is thus

$$\text{A.B. } (P_{\text{BIT}})_{\gamma_b \rightarrow 0} \approx Q(\sqrt{2\theta R \gamma_b}).$$

We note that if C is the binary $(n, 1)$ code, then $R = 1/n$, $\theta = n$ and

$$\text{A.B. } (P_{\text{BIT}})_{\gamma_b \rightarrow 0} \approx Q(\sqrt{2\gamma_b}),$$

which is the probability of bit error when no coding is used.

Also, we note that if the dual code of C has minimum Hamming distance greater than 2, then $\theta = 1$ and

$$\text{A.B. } (P_{\text{BIT}})_{\gamma_b \rightarrow 0} \approx Q(\sqrt{2R \gamma_b}),$$

which is the probability of bit error before decoding.

High SNR case

Since the all-0 code word is a member of S_0 ,

$$\sum_{\underline{c} \in S_0} \exp(-\underline{r} \cdot \underline{c} \sqrt{4R\gamma_b/N_0}) > 1$$

in (5). In the case $\underline{r} \cdot \underline{c} > 0$ for all $\underline{c} \in S_1$,

$$\sum_{\underline{c} \in S_1} \exp(-\underline{r} \cdot \underline{c} \sqrt{4R\gamma_b/N_0}) \approx 0$$

for large values of γ_b . Thus A.B. $(P_{BIT})_{\gamma_b \rightarrow \infty}$ is upper bounded by

$$A.B. (P_{BIT})_{\gamma_b \rightarrow \infty} \leq P[\cup_{\underline{c} \in S_1} \underline{r} \cdot \underline{c} \leq 0] \leq \sum_{\underline{c} \in S_1} P(\underline{r} \cdot \underline{c} \leq 0).$$

We now show that this upper bound is tight for sufficiently large values of γ_b , i.e.

$$A.B. (P_{BIT})_{\gamma_b \rightarrow \infty} \approx \sum_{\underline{c} \in S_1} P(\underline{r} \cdot \underline{c} \leq 0).$$

Let \underline{c}_1 and \underline{c}_2 be two code words of C . First note that $\underline{r} \cdot \underline{c}_1 = \underline{e} \cdot \underline{c}_1 + \sqrt{E}w(\underline{c}_1)$ and $\underline{r} \cdot \underline{c}_2 = \underline{e} \cdot \underline{c}_2 + \sqrt{E}w(\underline{c}_2)$, where $w(\underline{c})$ is the Hamming weight of \underline{c} . Without loss of generality, we assume that $w(\underline{c}_1) \leq w(\underline{c}_2)$. Let \underline{e}_1 be the solution to the problem of minimizing $\underline{e} \cdot \underline{e}$ subject to the constraints $\underline{e} \cdot \underline{c}_1 + \sqrt{E}w(\underline{c}_1) \leq 0$ and $\underline{e} \cdot \underline{c}_2 + \sqrt{E}w(\underline{c}_2) \leq 0$, and let \underline{e}_2 be the solution to the problem of minimizing $\underline{e} \cdot \underline{e}$ subject to $\underline{e} \cdot \underline{c}_1 + \sqrt{E}w(\underline{c}_1) \leq 0$. It is easy to show that $\underline{e}_1 \cdot \underline{e}_1 - \underline{e}_2 \cdot \underline{e}_2 = Ez$, $z > 0$, from which it follows that for sufficiently large values of E (and thus γ_b),

$$P(\underline{r} \cdot \underline{c}_1 \leq 0) \gg P(\underline{r} \cdot \underline{c}_1 \leq 0, \underline{r} \cdot \underline{c}_2 \leq 0). \quad (8)$$

But then

$$P\left[\bigcup_{\tilde{c} \in S_1} \tilde{r} \cdot \tilde{c} \leq 0\right] \approx \sum_{\tilde{c} \in S_1} P(\tilde{r} \cdot \tilde{c} \leq 0) ,$$

since, with high probability, only one of the inner products $\tilde{r} \cdot \tilde{c}$ will make a significant contribution. Again by (8), we may conclude that for sufficiently large values of γ_b ,

$$\text{A.B.}_{\gamma_b \rightarrow \infty} (P_{\text{BIT}}) \approx P\left(\bigcup_{\tilde{c} \in S_1} \tilde{r} \cdot \tilde{c} \leq 0\right) .$$

Thus

$$\text{A.B.}_{\gamma_b \rightarrow \infty} (P_{\text{BIT}}) \approx \sum_{\tilde{c} \in S_1} P(\tilde{r} \cdot \tilde{c} \leq 0) .$$

Now we know that

$$P(\tilde{r} \cdot \tilde{c} \leq 0) = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{2Rw(\tilde{c})\gamma_b}}^{\infty} \exp(-x^2/2) dx = Q(\sqrt{2Rw(\tilde{c})\gamma_b}) ,$$

from which it follows that

$$\text{A.B.}_{\gamma_b \rightarrow \infty} (P_{\text{BIT}}) \approx \sum_{\tilde{c} \in S_1} Q(\sqrt{2Rw(\tilde{c})\gamma_b}) .$$

Let w_m be the minimum weight of code words in S_1 and $N(w_m)$ the number of code words of weight w_m . Since only these code words make a significant contribution to the sum, the desired asymptotic expression for high SNR is seen to be

$$\text{A.B.}_{\gamma_b \rightarrow \infty} (P_{\text{BIT}}) = N(w_m) Q(\sqrt{2Rw_m\gamma_b}) .$$

If the code is cyclic with minimum distance d , then $w_m = d$ and

$$\text{A.B.}_{\gamma_b \rightarrow \infty} (P_{\text{BIT}}) \approx N(d) Q(\sqrt{2Rd\gamma_b}) .$$

Section 4

PARITY CHECK SET CONSTRUCTIONS

It is known⁽³⁹⁾ that any digital decoding function for a linear binary code can be realized as a weighted majority of nonorthogonal parity checks. An open question of practical interest is: For an (n,k) linear block code, how do we find a small subset of the 2^{n-k} available parity checks that is capable of correctly decoding the first received bit in spite of any pattern of t or fewer errors? In this note we present two approaches to constructing such subsets. The first approach, which applies to cyclic codes only, is based on "squaring", an automorphism of any binary cyclic code. The second approach, which is applicable to any linear binary code, is based on a "measure of reliability".

THE SQUARING APPROACH:

Let R_n be the ring of polynomials modulo x^n-1 over $GF(2)$. A cyclic code C of block length n is an ideal of R_n . The generator of C , $g(x)$, is a divisor of x^n-1 .

Let Π_2 denote the permutation $k \rightarrow 2k \pmod{n}$ of $\{0,1,\dots,n-1\}$. Π_2 induces the "squaring" automorphism of R_n

$$\Pi_2\left(\sum_{i=0}^{n-1} a_i x^i\right) = \sum_{i=0}^{n-1} a_i x^{2i}, \quad a_i \in GF(2).$$

Since the square of a multiple of $g(x)$ is also a multiple of $g(x)$, a binary cyclic code is invariant under the operation of squaring, and moreover the square of a code word of Hamming

weight w is another code word of weight w . The set of all code words obtained by squaring a code word \underline{v} is called the "square set of \underline{v} ".

Let E_1, \dots, E_ℓ denote the cycles of Π_2 and let v denote the set of position indices where the code word $\underline{v} \in C$ has ones. Define a function $f(\underline{v}) = (n_1, \dots, n_\ell)$, where $n_i = |E_i \cap v|$. It is easily shown⁽⁴³⁾ that $f(\underline{v}) = f(\underline{v}^2)$ for any $\underline{v} \in C$. A straightforward application of this property yields the following theorem on $M_{\underline{v}}$, the incidence matrix of the square set of \underline{v} :

Theorem:

Let

$$M_{\underline{v}} = \begin{bmatrix} \underline{v} \\ \underline{v}^2 \\ \vdots \\ \underline{v}^{2^{(m-1)}} \end{bmatrix}$$

be the incidence matrix of the square set of \underline{v} and suppose $f(\underline{v}) = (n_1, \dots, n_\ell)$, where $n_j = |E_j \cap v|$. Then the columns of $M_{\underline{v}}$ corresponding to the components of E_j each contain exactly $\frac{mn_j}{|E_j|}$ 1's, where m is the multiplicative order of 2 (mod n).

Example 1

Suppose $n = 7$. Then $E_1 = \{0\}$, $E_2 = \{1, 2, 4\}$, $E_3 = \{3, 6, 5\}$. Let C be the $(7, 6)$ code and $v = \{0, 1\}$. Then $v^2 = \{0, 2\}$, $v^4 = \{0, 4\}$, $v^8 = v$ and $f(\underline{v}) = f(\underline{v}^2) = f(\underline{v}^4) = (1, 1, 0)$. The incidence matrix

of the square set of \underline{v} is

$$M_v = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} & . \end{matrix}$$

Now let C be the dual of the code we wish to decode and let \underline{v} and \underline{v}' be code words of C with '1' in the first position. Assume $f(\underline{v}) = (1, n_2, \dots, n_\ell)$ and $f(\underline{v}') = (1, n'_2, \dots, n'_\ell)$ and that $f(\underline{v}) \neq f(\underline{v}')$. If it is possible to find integers $a > 0$ and $b > 0$ such that $an_j + bn'_j \leq \lambda$ for $j = 2, \dots, \ell$, then the parity checks of the square set of \underline{v} , replicated a times, can be combined with the parity checks of the square set of \underline{v}' , replicated b times, to obtain a nonorthogonal parity check set in which the first error position e_0 is checked by all of the parity checks, but no other position is checked by more than λ parity checks.

Example 2 (17,9) code

The (17,9) double-error-correcting quadratic residue code is not L -step orthogonalizable. However, this code can be weighted-majority decoded in one step using 15 nonorthogonal parity checks as we now show.

For $n = 17$, $E_1 = \{0\}$, $E_2 = \{1, 2, 4, 8, 16, 15, 13, 9\}$ and $E_3 = \{3, 6, 12, 7, 14, 11, 5, 10\}$. Two code words of weight 6 in the (17,8) dual code for which $f(v) \neq f(v')$ are:

$$\begin{aligned} v &= \{0, 1, 3, 6, 8, 9\} & f(v) &= (1, 3, 2) \\ v' &= \{0, 4, 5, 6, 7, 11\} & f(v') &= (1, 1, 4) . \end{aligned}$$

If 'a' is the weight (number of replications) assigned to the square set of v and 'b' is the weight assigned to the square set of v', then the following equations must be solved for a, b, and λ :

$$3a + b = \lambda$$

$$2a + 4b = \lambda \quad .$$

The simplest solution is $a = 3$, $b = 1$, $\lambda = 10$. There are 8 words in each square set and each word checks the first position. Applying Ng's bound⁽⁴⁴⁾ with $r = 3(8) + 1(8) = 32$ and $\lambda = 10$, we see that it is possible to decode the first received bit in spite of any pattern of $t = \frac{r+\lambda-1}{2\lambda} = 2$ or fewer errors. This nonorthogonal parity check set and its associated weights are shown in Table 1. Note that since the minimum weight of any syndrome of an error pattern with $e_0 = 1$ is 2 greater than the maximum weight of any syndrome of an error pattern with $e_0 = 0$, the last check may be discarded, thereby reducing the number of nonzero parity checks required to 15.

THE MEASURE OF RELIABILITY APPROACH:

Another approach to selecting a parity check set is to define a measure of the "reliability" of a parity check and then use this measure to select a subset of the 2^{n-k} checks available. One such measure of reliability is the absolute value of the number of times a parity check is "right" minus the number of times it is "wrong" over the set of error patterns of interest, where we say that a parity check is "right" for an error pattern $e = (e_0, \dots, e_{n-1})$ if the check sum is equal to e_0 ; otherwise it

is "wrong". The error patterns of interest in the examples to follow are those of weight t or less, where t is the guaranteed error-correction capability of the code. The general idea is to use the parity checks with the highest reliability coefficients. When a parity check is "wrong" more often than it is "right", we usually use the complement of the check (denoted by a minus sign assigned to the weight).

<u>Word in the (17,8) Dual Code</u>	<u>Assigned Weight</u>
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	10
1 1 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0	3
1 1 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1	3
1 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0	3
1 0 0 0 1 0 0 1 1 0 0 0 0 0 1 1 0 0	3
1 0 0 0 0 0 0 0 0 1 1 0 1 0 0 1 0 1	3
1 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1	3
1 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0	3
1 0 0 1 1 0 0 0 0 0 1 1 0 0 1 0 0 0	3
1 0 0 0 1 1 1 1 0 0 0 0 1 0 0 0 0 0	1
1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0 0	1
1 0 0 1 0 0 0 1 0 0 1 1 0 0 0 0 0 1	1
1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 1 0	1
1 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0	1
1 0 0 1 0 1 0 1 0 1 0 0 1 0 0 0 0 0	1
1 1 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0	1
1 0 1 1 0 0 0 0 0 0 0 0 1 1 0 1 0 0	1

Table 1. Parity check set used to decode the (17,9) code

Example 3 (15,5) code

The (15,5) triple-error-correcting cyclic Reed-Muller code can be majority decoded in two steps using 42 orthogonal parity

checks^(22,29). We now show that this code can be weighted majority decoded in one step using 31 nonorthogonal parity checks.

The (15,10) dual code has the following weight distribution:

Code word weight	0	4	6	8	10	12
Number of code words	1	105	280	435	168	35

The error patterns of interest in this case are the patterns of 3 or fewer errors, and the reliability coefficients of the parity checks with respect to this set of error patterns are as given in Table 2. The zero parity check is the most reliable, the weight-12 parity checks which do not check e_0 are the second most reliable, and the weight-4 parity checks which check e_0 are the third most reliable. (We use the complements of the second set of check sums since they are "wrong" more often than they are "right".) The nonorthogonal parity check set formed from these three sets of checks and the associated weights are shown in Table 3. (A negative weight indicates that the complement of the check sum is to be used.) It is possible to discard the first four nonzero checks, thereby reducing the required number of nonzero parity checks to 31.

	<u>Code Word Weight</u>	<u>No. of Words of That Weight</u>	<u>Reliability Coefficient</u>
parity checks which check e_0	4	28	$ 99 = 99$
	6	112	$ -5 = 5$
	8	232	$ -13 = 13$
	10	112	$ 11 = 11$
	12	28	$ 3 = 3$
parity checks which do not check e_0	0	1	$ 363 = 363$
	4	77	$ -5 = 5$
	6	168	$ -13 = 13$
	8	203	$ 11 = 11$
	10	56	$ 3 = 3$
	12	7	$ -101 = 101$

Table 2. Reliability coefficients for words in the (15,10) dual code

<u>Words in the (15,10) Dual Code</u>	<u>Assigned Weight</u>
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	6
1 1 1 0 0 0 0 0 0 0 1 0 0 0 0	1
1 1 0 1 0 0 0 0 1 0 0 0 0 0 0	1
1 1 0 0 0 1 0 0 1 0 0 0 0 0 0	1
1 1 0 0 0 0 1 0 0 0 0 0 1 0 0	1
1 1 0 0 0 0 0 0 0 1 0 0 0 0 1	1
1 1 0 0 0 0 0 0 0 0 0 1 0 1 0	1
1 0 1 1 0 0 0 0 0 0 0 0 0 1 0	1
1 0 1 0 1 1 0 0 0 0 0 0 0 0 0	1
1 0 1 0 0 0 1 0 0 0 0 0 0 0 1	1
1 0 1 0 0 0 0 1 0 0 0 1 0 0 0	1
1 0 1 0 0 0 0 0 0 1 0 0 1 0 0	1
1 0 0 1 1 0 0 0 0 1 0 0 0 0 0	1
1 0 0 1 0 1 0 0 0 0 0 0 1 0 0	1
1 0 0 1 0 0 1 0 1 0 0 0 0 0 0	1
1 0 0 1 0 0 0 0 0 0 1 1 0 0 0	1
1 0 0 0 1 0 1 0 0 0 0 1 0 0 0	1
1 0 0 0 1 0 0 1 0 0 0 0 0 0 1	1
1 0 0 0 1 0 0 0 1 0 1 0 0 0 0	1
1 0 0 0 1 0 0 0 0 0 0 0 1 1 0	1
1 0 0 0 0 1 1 1 0 0 0 0 0 0 0	1
1 0 0 0 0 1 0 0 0 1 0 0 0 1 0	1
1 0 0 0 0 1 0 0 0 0 0 1 0 0 1	1
1 0 0 0 0 0 1 0 0 1 1 0 0 0 0	1
1 0 0 0 0 0 0 1 1 0 0 0 1 0 0	1
1 0 0 0 0 0 0 1 0 0 1 0 0 1 0	1
1 0 0 0 0 0 0 0 1 1 0 1 0 0 0	1
1 0 0 0 0 0 0 0 1 0 0 0 0 1 1	1
1 0 0 0 0 0 0 0 0 0 1 0 1 0 1	1
0 0 1 1 0 1 1 1 1 1 1 1 1 1 1	-1
0 1 0 1 1 1 1 1 1 0 1 1 1 1 1	-1
0 1 1 0 1 1 1 1 1 1 1 1 1 1 0	-1
0 1 1 1 1 0 1 1 1 1 1 0 1 1 1	-1
0 1 1 1 1 1 0 1 1 1 1 1 1 0 1	-1
0 1 1 1 1 1 1 0 1 0 1 1 1 1 1	-1
0 1 1 1 1 1 1 1 1 1 0 0 1 1 1	-1

Table 3. Parity check set used to decode the (15,5) code

Section 5

CONCLUSIONS

As pointed out in Section 2, we are now in a position to translate all of the digital decoding algorithms of classical algebraic coding theory into analog decoding algorithms via the isomorphism between the additive S -domain and the multiplicative S' -domain. There are some enticing experiments begging to be conducted. For example, how well would a multiplicative BCH decoding algorithm perform when applied to an unquantized received word? How are we to interpret such a decoding algorithm? The very concept of BCH codes and their decoding algorithms is based on the assumption that a digital error vector of no more than t nonzero components has been added to the code word in the finite-field domain of the code. From the viewpoint of classical algebraic coding theory, it makes no sense to talk about error vectors with analog components. Yet we know that a multiplicative BCH decoding algorithm will correct some set of error vectors. The only question is: what set? (Actually, it is probable that a raw, unmodified multiplicative BCH decoder would decode only those digital errors that it would have decoded in the additive domain, since a BCH decoder (unlike a threshold decoder) may elect not to decode a received word. But once in the multiplicative domain, a great many "loosening up" modifications suggest themselves. One might consider generalizing the idea of "root", for instance.)

It is also quite possible that there exists an analog decoding algorithm (and a new class of codes, perhaps?) which is analogous to BCH decoding, but which is designed to correct only those error vectors which fall within a continuous sphere of Euclidean radius t , rather than being designed to correct only those digital error vectors which fall within a discrete "sphere" of Hamming radius t . Of course, the ability to derive analog decoding methods which are analogous to, rather than direct translations of, classical digital decoding algorithms requires an understanding of the principles of decoding in the multiplicative domain which we do not yet possess.

In addition to new analog decoding algorithms obtained by direct translation of existing digital algorithms, or constructed by analogy with existing digital algorithms, there is the possibility of devising entirely new decoding methods which have no counterpart in classical algebraic coding theory. This possibility stems from the new algebraic properties acquired when we move from the classical additive domain to the multiplicative domain. For example, in this new domain a digital decoding function can always be extended to an analytic function. This means that all of the techniques of real and complex analysis become available. One thinks immediately of hill-climbing techniques which will, with the unquantized received word as the starting point, converge to the nearest code word. We have in fact tried this, and our first experiment with a convergence technique had an outcome which we should have been able to

predict.

In the course of our work on the optimum symbol-by-symbol decoding algorithm described in the previous section, we asked ourselves how we could modify the algorithm to do optimum word decoding (and thus be equivalent to correlation and Viterbi decoding). We designed an experiment for the white Gaussian channel in which we fed back the unquantized output of the optimum symbol-by-symbol decoder to the input (after the initial processing of the unquantized received word) and then iterated the process until it stabilized. To our delight, the contents of the fed-back decoder did indeed converge to the nearest code word. We then became interested in the speed of convergence and found that by "fooling" the decoder* into thinking that the SNR on the channel was higher than it actually was, the rate of convergence was increased. In the end we found that by providing the decoder with a sufficiently high artificial SNR, the nearest code word was always produced on the first pass. In other words, iteration was not necessary! In retrospect, it is obvious that this should be the case. After all, one need not derive the SNR for correlation or Viterbi decoding, and besides, the optimum symbol-by-symbol decoding algorithm uses all of the available parity checks, which should have led us to suspect

*"Fooling" the decoder consists of using an artificially high or low SNR when computing the likelihood ratio. This has the effect of changing the shape of the (adaptive) soft-decision decoding function.

that nothing further would be gained by iteration.

What this result does suggest is that there should be a way to trade off speed of convergence and hardware complexity. If all of the parity checks are used, the hardware component of complexity is maximized and the time component is minimized. If too few parity checks are used, the decoder does not converge to the nearest code word, and in fact may not converge to a code word at all. Based on experience in other areas where convergence techniques are widely used, we expect to find an operating range in which the hardware and time components of complexity can be traded off (with a further trade-off involving error probability), and that the optimum operating point for most applications will involve iteration. We consider this line of investigation to be one of the most exciting in the period ahead. We might note here that a similar idea has been suggested by Chase⁽⁴⁵⁾ who plans to investigate a "cascade" decoder (a soft-decision decoder following by a hard-decision decoder).

In Section 3, we presented a symbol-by-symbol decoding rule for linear codes which is optimum in the sense that it minimizes the probability of symbol error on a time-discrete memoryless channel when the code words are equiprobable. A comment or two on the relationship between this technique and correlation/Viterbi decoding would seem to be in order.

First, although the performance of correlation/Viterbi decoding is inferior to the performance of the decoding rule

presented here on a symbol-error basis, and vice versa on a word-error basis, some preliminary simulation results for the white Gaussian channel suggest that the two approaches are very close in performance on either basis. Symbol-error-rate is generally considered to be a better measure of performance than word-error-rate, especially in the case of convolutional codes, and this would seem to give a slight edge to the decoding rule presented here. On the other hand, correlation/Viterbi decoding is applicable to nonlinear as well as linear codes, which might be an advantage in some applications. Our present feeling is that for all practical purposes the two approaches give virtually the same performance.

When we turn to the question of complexity, however, there is a considerable difference between the two decoding techniques. Correlation/Viterbi decoding is only practical for low rate or short codes whereas the symbol-by-symbol decoding rule is only practical for high rate or short linear codes. We are fairly well convinced, and the reader may be able to convince himself by studying the examples in Section 3.2, that the complexity of the symbol-by-symbol decoding rule for an (n,k) linear code is comparable to the complexity of a correlation/Viterbi decoder for the $(n,n-k)$ dual code. This is fairly easy to see in the case of linear block codes, but not so obvious in the case of convolutional codes since there are so many options and programming tricks to be considered. The authors, however, are firm believers in the coding-complexity Folk Theorem: "The

complexity of any function defined on a linear code is comparable to the complexity of (essentially) that same function defined on the dual code". (In fact, it was the unsatisfying lack of a soft decision decoding method for high rate linear codes that was "dual" to correlation/Viterbi decoding that motivated the research reported here.) If our intuition is correct, then the symbol-by-symbol decoding rule and correlation/Viterbi decoding should be of comparable complexity for rate $1/2$ codes. We remark that the decoding rule for linear codes over $GF(p)$ can be generalized in a straightforward fashion to linear codes over $GF(p^m)$ by using the generalized finite Fourier transform of [46, pg. 367].

For very high SNR, the asymptotic expression for bit error probability derived in Section 3.3 is the same whether optimum bit-by-bit decoding or maximum-likelihood word decoding (i.e. correlation) is used. The reason for this is easily seen intuitively. For $\gamma_b \rightarrow \infty$, with high probability the only time a decoding error occurs using either scheme is when the received word lies very nearly on a straight line between the transmitted code word and a "nearest neighbor" code word, slightly closer to the neighbor. In either case, the pattern of errors coincides with the positions in which the transmitted code word and the neighbor differ. This result tends to support the conjecture⁽¹³⁾ that correlation decoding and optimum symbol-by-symbol decoding give, for all practical purposes, the same

performance on discrete memoryless channels.

Finally, we conjecture that the one-step weighted majority decoders for the $(17,9)$ and $(15,5)$ codes derived in Section 4 to illustrate the two parity set construction methods are in fact minimal decoders in the sense that they use the fewest possible parity checks for t -error-correction. It is interesting to note: (1) that in the $(17,9)$ decoder, parity checks of equal reliability are assigned different weights, and (2) that it is apparently necessary to use the complements of consistently "wrong" parity check sums to obtain a minimal decoder for the $(15,5)$ code. It is our feeling, however, that these are "quirks" related to digital t -error-correction, and would probably not carry over to soft-decision decoding.

REFERENCES

1. Hartmann, C. R. P. and L. D. Rudolph, "Generalized Finite-Geometry Codes," Proceedings of the Tenth Annual Allerton Conference on Circuit and System Theory, University of Illinois, Urbana, Ill., October, 1972.
2. Ducey, J. B., C. R. P. Hartmann, and L. D. Rudolph, "Some Results on Generalized Projective-Geometry Codes," presented at the IEEE International Symposium on Info. Theory, Ashkelon, Israel, June 1973.
3. Hartmann, C. R. P., J. B. Ducey and L. D. Rudolph, "On the Structure of Generalized Finite-Geometry Codes," IEEE Trans. on Info. Theory, IT-20, 240-252 (1974).
4. Rudolph, L. D. and C. R. P. Hartmann, "Decoding by Sequential Code Reduction," presented at the IEEE International Symposium on Information Theory, Asilomar, Calif. (1972) and published in the IEEE Trans. on Info. Theory, IT-19, 549-555 (1973).
5. Rudolph, L. D. and C. R. P. Hartmann, "Practical Implementation of Very Long Cyclic Codes," presented at the Canadian Communication Conference, Montreal, November 1972.
6. Rudolph, L. D., "Some Current Research in Decoding Theory," invited lectures given at the Centre International des Sciences Mechaniques, Udine, Italy, July 1974. To be published in Coding and Complexity.
7. Riek, J. R., C. R. P. Hartmann and L. D. Rudolph, "Majority Decoding of Some Classes of Binary Cyclic Codes," presented at the IEEE International Symposium on Information Theory, Notre Dame, Ind., October 1974, and published in the IEEE Trans. on Info. Theory, IT-20, 637-643 (1974).
8. Rudolph, L. D. and C. R. P. Hartmann, "Maximum-Radius Analog Threshold Decoding," presented at the IEEE International Symposium on Info. Theory, Notre Dame, Ind., October 1974.
9. Rudolph, L. D., "Integrated Decoding-Demodulation," invited address before the Syracuse Chapter of the IEEE Information Theory Group, Syracuse, N. Y., January 1975.
10. Rudolph, L. D., "Algebraic Analog Decoding: a new Setting for old ideas," invited talk presented at the IEEE Information Theory Workshop, Lenox, Mass., June 1975.

11. Rudolph, L. D., "Some Recent Results in Analog Decoding," invited talk, Electrical Engineering Department Colloquium, University of Waterloo, October 1975.
12. Rudolph, L. D., "Soft-Decision Symbol-by-Symbol Decoding in the Dual Code Domain," invited talk at the IEEE Communication Theory Workshop, Captiva, Florida, April 1976.
13. Hartmann, C. R. P. and L. D. Rudolph, "An Optimum Symbol-by-Symbol Decoding Rule for Linear Codes," presented at the IEEE International Symposium on Information Theory, Ronneby, Sweden, June 1976. Also published in the IEEE Transactions on Information Theory, Vol. IT-22, September 1976.
14. Hartmann, C. R. P., L. D. Rudolph and K. G. Mehrotra, "Asymptotic Performance of Optimum Bit-by-Bit Decoding for the White Gaussian Channel," School of Computer and Information Science Technical Report 2-76, Syracuse University, June 1976. Submitted for publication in the IEEE Transactions on Information Theory.
15. Levinson, N., "Coding Theory: A Counterexample to G. H. Hardy's Conception of Applied Mathematics," Amer. Math. Monthly, Vol. 77, No. 3, pp. 249-258, March 1970.
16. Massey, J. L., "Coding and Demodulation in Digital Communications," Proceedings of the International Zurich Seminar on Digital Communications, Switzerland, March 1974.
17. Viterbi, A. J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. Info. Theory, Vol. IT-13, pp. 260-269, April 1967.
18. Silverman, R. A. and M. Balser, "Coding for Constant-Data-Rate System," Proc. IRE, Vol. 42, pp. 1428-1435, Sept. 1954 and Proc. IRE, Vol. 43, pp. 728-733, June 1955.
19. Forney, C. D., Jr., "Generalized Minimum Distance Decoding," IEEE Trans. Info. Theory, Vol. IT-12, pp. 125-131, April 1966.
20. Weldon, E. J., Jr., "Decoding Binary Block Codes on Q-ary Output Channels," IEEE Trans. Info. Theory, Vol. IT-17, pp. 713-718, November 1971.
21. Chase, D., "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," IEEE Trans. Info. Theory, Vol. IT-18, pp. 170-182, January 1972.

22. Reed, I. S., "A Class of Multiple Error-Correcting Codes and the Decoding Scheme," IRE Trans., PGIT-4, 38-49 (1954).
23. Muller, D. E., "Application of Boolean Algebra to Switching Circuit Design and to Error Detection," IRE Trans., EC-3, 6-12 (1954).
24. Yale, R. B., "Error Correcting Codes and Linear Recurring Sequences," Lincoln Laboratory Group Report 34-77, M.I.T. Lincoln Labs. (1958).
25. Zierler, N., "On a Variation of the First Order Reed-Muller Codes," Lincoln Laboratory Group Report 34-88, M.I.T. Lincoln Labs. (1958).
26. Green, J. H., and R. L. San Soucie, "An Error-Correcting Encoder and Decoder of High Efficiency," Proc. IRE, 46, 1741-1744 (1958).
27. Prange, E., "The Use of Coset Equivalence in the Analysis and Decoding of Group Codes," AFCRL-TR-59-164, Air Force Cambridge Research Center, Cambridge, Mass. (1959).
28. Mitchell, M. E. et. al., "Coding and Decoding Operations Research," General Electric Co. Final Report on Contract No. AF 19(604)-6183 for Air Force Cambridge Research Center (1961).
29. Massey, J. L., Threshold Decoding, M.I.T. Press (1963).
30. Rudolph, L. D., "Geometric Configurations and Majority Logic Decodable Codes," MEE Thesis, University of Oklahoma, Norman, Oklahoma (1964).
31. Rudolph, L. D., "A Class of Majority Logic Decodable Codes," IEEE Trans. on Info. Theory, IT-13, pp. 305-307 (1967).
32. Chen, D. L., "Note on Majority-Logic Decoding of Finite Geometry Codes," IEEE Trans. on Info. Theory, IT-18, pp. 539-541 (1972).
33. Duc, N. Q., "Pseudostep Orthogonalization: A New Threshold-Decoding Algorithm," IEEE Trans. on Info. Theory, IT-17, pp. 766-768 (1971).
34. Gore, W. C., "Generalized Threshold Decoding and the Reed-Solomon Codes," IEEE Trans. on Info. Theory, IT-15, pp. 78-81 (1969).

35. Kasami, T. and S. Lin, "On Majority-Logic Decoding for the Duals of Primitive Polynomial Codes," IEEE Trans. on Info. Theory, IT-17, pp. 322-331 (1971).
36. Ng, S. W., "On Rudolph's Majority Logic Decoding Algorithm," IEEE Trans. on Info. Theory, IT-16, p. 651 (1970).
37. Rudolph, L. D., "Threshold Decoding of Cyclic Codes," IEEE Trans. on Info. Theory, IT-15, pp. 414-418 (1969).
38. Rudolph, L. D., "Generalized Threshold Decoding of Convolutional Codes," IEEE Trans. on Info. Theory, IT-16, pp. 739-745 (1970).
39. Rudolph, L. d. and Robbins, W. E., "One-Step Weighted-Majority Decoding," IEEE Trans. on Info. Theory, IT-18, pp. 446-448 (1972).
40. Weldon, E. J., Jr., "Some Results on Majority-Logic Decoding," Chapter 8, Error Correcting Codes, H. Mann, Ed., Wiley (1968).
41. CNR, Inc., R & D Status Report No. 2 on Contract F30602-76-C-0361, Sept. 1976.
42. Peterson, W. W. and E. J. Weldon, Jr., "Error-Correcting Codes," (2nd Ed.), M.I.T. Press (1972).
43. Longobardi, R. J., "Some Results on Majority Decoding of Binary Cyclic Codes," Ph.D. Dissertation, Systems and Information Science, Syracuse University, Syracuse, N. Y., Sept. 1975.
44. Ng, S. W., "On Rudolph's Majority-Logic Decoding Algorithm," IEEE Trans. on Info. Theory, Vol. IT-16, pp. 651-652, Sept. 1970.
45. CNR, Inc., R & D Status Report No. 4 on Contract F30602-76-C-0361, November 1976.
46. Assmus, E. F., Jr., H. F. Mattson, Jr., "Coding and Combinatorics," SIAM Review, Vol. 16, No. 3, pp. 349-388, July 1974.

METRIC SYSTEM

BASE UNITS:

Quantity	Unit	SI Symbol	Formula
length	metre	m	...
mass	kilogram	kg	...
time	second	s	...
electric current	ampere	A	...
thermodynamic temperature	kelvin	K	...
amount of substance	mole	mol	...
luminous intensity	candela	cd	...

SUPPLEMENTARY UNITS:

plane angle	radian	rad	...
solid angle	steradian	sr	...

DERIVED UNITS:

Acceleration	metre per second squared	...	m/s
activity (of a radioactive source)	disintegration per second	...	(disintegration)/s
angular acceleration	radian per second squared	...	rad/s
angular velocity	radian per second	...	rad/s
area	square metre	...	m
density	kilogram per cubic metre	...	kg/m
electric capacitance	farad	F	A·s/V
electrical conductance	siemens	S	A/V
electric field strength	volt per metre	...	V/m
electric inductance	henry	H	V·s/A
electric potential difference	volt	V	W/A
electric resistance	ohm	...	V/A
electromotive force	volt	V	W/A
energy	joule	J	N·m
entropy	joule per kelvin	...	J/K
force	newton	N	kg·m/s
frequency	hertz	Hz	(cycle)/s
illuminance	lux	lx	lm/m
luminance	candela per square metre	...	cd/m
luminous flux	lumen	lm	cd·sr
magnetic field strength	ampere per metre	...	A/m
magnetic flux	weber	Wb	V·s
magnetic flux density	tesla	T	Wb/m
magnetomotive force	ampere	A	...
power	watt	W	J/s
pressure	pascal	Pa	N/m
quantity of electricity	coulomb	C	A·s
quantity of heat	joule	J	N·m
radiant intensity	watt per steradian	...	W/sr
specific heat	joule per kilogram-kelvin	...	J/kg·K
stress	pascal	Pa	N/m
thermal conductivity	watt per metre-kelvin	...	W/m·K
velocity	metre per second	...	m/s
viscosity, dynamic	pascal-second	...	Pa·s
viscosity, kinematic	square metre per second	...	m/s
voltage	volt	V	W/A
volume	cubic metre	...	m
wavenumber	reciprocal metre	...	(wave)/m
work	joule	J	N·m

SI PREFIXES:

Multiplication Factors	Prefix	SI Symbol
1 000 000 000 000 = 10 ¹²	tera	T
1 000 000 000 = 10 ⁹	giga	G
1 000 000 = 10 ⁶	mega	M
1 000 = 10 ³	kilo	k
100 = 10 ²	hecto*	h
10 = 10 ¹	deka*	da
0.1 = 10 ⁻¹	deci*	d
0.01 = 10 ⁻²	centi*	c
0.001 = 10 ⁻³	milli	m
0.000 001 = 10 ⁻⁶	micro	μ
0.000 000 001 = 10 ⁻⁹	nano	n
0.000 000 000 001 = 10 ⁻¹²	pico	p
0.000 000 000 000 001 = 10 ⁻¹⁵	femto	f
0.000 000 000 000 000 001 = 10 ⁻¹⁸	atto	a

* To be avoided where possible.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C³) activities, and in the C³ areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

